British Machine Vision Conference (BMVC) 2012

Towards Longer Long-Range Motion Trajectories

Michael Rubinstein[1]
mrub@mit.edu
Ce Liu[2]
celiu@microsoft.com
William T. Freeman[1]
billf@mit.edu

[1] MIT CSAIL
[2] Microsoft Research New England

Abstract

Although dense, long-range, motion trajectories are a prominent representation of motion in videos, there is still no good solution for constructing dense motion tracks in a truly long-range fashion. Ideally, we would want every scene feature that appears in multiple, not necessarily contiguous, parts of the sequence to be associated with the same motion track. Despite this reasonable and clearly stated objective, there has been surprisingly little work on general-purpose algorithms that can accomplish this task. State-of-the-art dense motion trackers process the sequence incrementally in a frame-by-frame manner, and associate, by design, features that disappear and reappear in the video, with different tracks, thereby losing important information of the long-term motion signal. In this paper, we strive towards an algorithm for producing generic long-range motion trajectories that are robust to occlusion, deformation and camera motion. We leverage accurate *local* (short-range) trajectories produced by current motion tracking methods and use them as an initial estimate for a *global* (long-range) solution. Our algorithm re-correlates the short trajectories and links them to form a long-range motion representation by formulating a combinatorial assignment problem that is defined and optimized globally over the entire sequence. This allows to correlate features in arbitrarily distinct parts of the sequence, as well as handle tracking ambiguities by spatiotemporal regularization. We report the results of the algorithm on both synthetic and natural videos, and evaluate the long-range motion representation for action recognition.

1 Introduction

There are two popular representations to characterize motion in videos: sparse feature point tracking and dense optical flow. In the first representation, (good) features are detected in one frame, and tracked independently in the rest of the frames [18], while in the latter representation, a flow vector is estimated for every pixel, indicating where the pixel moves to in the next frame [7, 13]. Fig. 1(a,b) illustrate these two representations in spatial-temporal domain. As revealed in [16], sparse feature point tracking can establish long-range correspondences (*e.g.* between hundreds of frames), but only a few feature points are detected. While useful for some applications, it is a very incomplete representation of the motion in a scene. For example, it is hard to infer important object information, such as shape, from a set of sparse feature points. On the other hand, dense optical flow reveals more about the moving objects, but the integer-grid-based flow fields cannot reliably propagate to faraway frames.

A natural solution, therefore, is to combine feature point tracking and dense optical flow fields to a set of *spatially dense* and *temporally smooth* trajectories (or particles, tracks) [16],

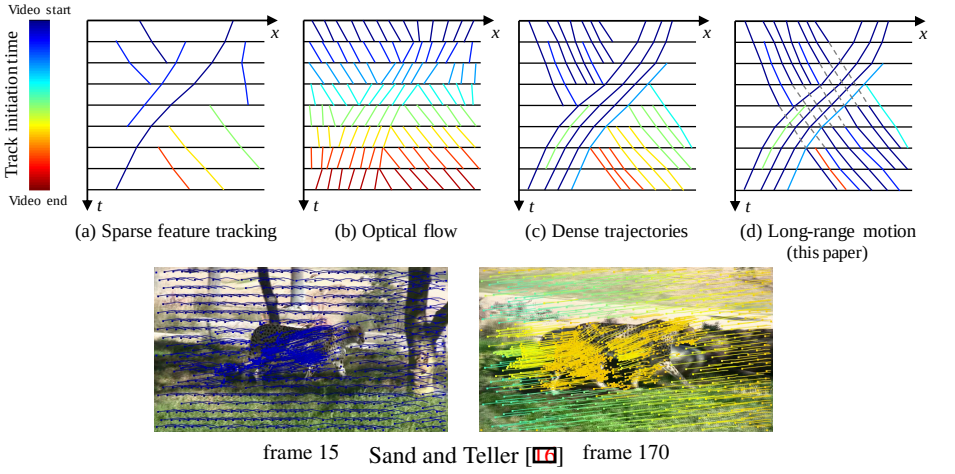


Figure 1: **Long-range motion vs. state-of-the-art.** Top: comparison on a canonical sequence. Bottom: typical results by a state-of-the-art dense point tracker [16] at two distinct frames of the *cheetah* sequence. The tracks are overlayed on the frames together with their recent path, colored by their initiation time in the sequence (top left). Distinct sets of motion tracks are covering the same main object five seconds apart in the video, due to occlusions, deformation and camera motion, thereby loosing important information on the long-term motion signal.

as shown in Fig. 1(c). Despite recent advances in obtaining dense trajectories from a video sequence [3, 21], it is challenging to obtain *long-range* dense trajectories. Consider, for example, the video sequence shown in Fig. 1. Representative frames from the source video are shown together with the motion tracks produced by Sand and Teller [16]. In two distant frames, the feature points on the same object have different colors, indicating that tracks for some physical points have disappeared and new ones were assigned, possibly due to occlusion, mis-tracking or camera motion. Therefore, important long-range correspondences for characterizing the motion in the scene are lost.

Most prior work on dense motion trajectories share a common framework: motion tracks are constructed based on the pairwise motion estimated from consecutive frames [16, 21, 22]. [21], for example, explicitly terminates tracks when occlusions are encountered, while the pixels may become visible again in later frames and will be assigned to new tracks. Particle video [16] takes a step towards a more global solution by sweeping the video forward and backward, but particles are still propagated from one frame to the next and higher-order correlations between frames are only considered, to some extent, within a single contiguous track. Other local attempts to handle occlusion at the feature level have been made, which are either restricted to particular scenes involving convex objects [15], or are limited in their ability to bridge over long occlusions due to their online nature [20].

In this paper, we propose a novel divide and conquer approach to long-range motion estimation. Given a long video or image sequence, we first produce high-accuracy *local* track estimates, or *tracklets*, and later propagate them into a *global* solution, while incorporating information from throughout the video. The tracklets are computed using state-of-the-art dense motion trackers that have become quite accurate for short sequences as demonstrated by standard evaluations [1]. Our algorithm then constructs the long-range tracks by linking the short tracks in an optimal manner. This induces a combinatorial matching problem that we solve simultaneously for all tracklets in the sequence.

Our method is inspired by the abundant literature on multi-target tracking, which deals with data association at the *object level* [2, 5, 6, 8, 10, 11, 14, 19]. Tracking objects and tracking pixels, however, are quite different in nature, for several reasons. First, many object

tracking methods are tightly coupled with the object detection algorithm or the application at hand, using particular image cues and domain-specific knowledge. In contrast, dense motion trajectories are defined at the *pixel level*, using generic low-level image features. Second, while there are typically few (say tens of) objects in a frame, there are billions of pixels, and millions of tracks to process in a video, which has implications on the algorithm formulation and design. Third, evaluating dense motion trackers is significantly more challenging than object tracking. Evidently, there exists numerous datasets for object tracking evaluation, yet, to our knowledge, there does not exist any dataset for evaluating dense long-range motion tracks. The novelty of our work is in pushing ideas from event and object linking down to the feature level, and our goal is to advance the state-of-the-art in all the above points.

The main contributions of this paper are: (a) a novel divide-and-conquer style algorithm for constructing dense, long-range motion tracks from a single monocular video, and (b) novel criteria for evaluating dense long-range tracking results with and without ground-truth motion trajectory data. We evaluate our approach on a set of synthetic and natural videos, and explore the utilization of long-range tracks for action recognition.

2 Long-range Motion Trajectories

The input to our system is a monocular video sequence I of T frames. The basic primitive in our formulation is a track, $\tau = \mathbf{x}(t)$, where $\mathbf{x}(t) = (x(t), y(t))$ are the track's spatial coordinates at time t. Also denote t^{start} and t^{end} the start and end time of track τ, respectively, and $\mathbf{x}(t) = \emptyset$ if τ is occluded at time t. The temporal coordinates t are always integral (frames), while the spatial coordinates $\mathbf{x}(t)$ are in sub-pixel accuracy. We denote by $\Omega = \{\tau_i\}$ the set of tracks in the video.

A possible set of tracks Ω to describe motion in the sequence is the one produced by pairwise optical flow, *i.e.* tracks of length 2 between consecutive frames. Similarly, it is always possible to add an additional track to describe the motion in some part of the sequence. Such short-range representations, however, do not model important characteristics of the motion signal over time. We define the long-range tracking problem as the problem of "covering" the scene with the minimal number of tracks such that each scene point is associated with exactly one track. This representation will result in a temporally-sparse set of tracks, with fewer long tracks as opposed to many short ones (Fig. 1(d)).

In this work, we take a divide-and-conquer approach to long-range tracking, solving first for short track segments, or *tracklets*, and later combining them to form long stable trajectories. Tracklets are estimated using state-of-the-art motion trackers, and we assume they are sufficiently dense so that approximately every scene feature is covered by some track. In this work, the spatial density of our representation will be subject to that of the tracking method we use for initialization, and we focus on the temporal density of the representation – minimizing the number of tracks covering a single scene feature – leaving treatment of the spatial density for future work.

3 The Algorithm

3.1 Initialization

We have experimented with several trackers, namely KLT [18], Particle Video (PV) [16], and the point tracker by Sundaram et al. (LDOF) [21], based on the large displacement optical flow method of [3]. PV and LDOF produce spatially-denser trajectories than KLT and are currently considered state-of-the-art in the field. In our experiments, LDOF consistently produced more plausible and stable tracklets, and so we chose it as initialization for our algorithm (we compare with initialization using PV in Sect. 4). We use the authors' implementation available online, and run their tracker using dense sampling (2×2 grid).

3.2 Track linking

Features that disappear and reappear will be assigned to different tracks, and so our goal is to combine them into long trajectories such that each scene feature is associated with a single track with high probability. This induces a combinatorial matching problem that we define and solve simultaneously for *all* tracklets in the sequence.

For each track that terminates within the sequence, we consider tracks spawned after its termination as possible continuing tracks. We call a track we would like to merge with another – *query* track – and tracks we consider to append it – *candidates*. Notice that a query track might itself be a candidate for another query track.

In a good association of candidates to queries, we expect (a) linked tracks to encode the same scene feature with high probability, (b) each query track and candidate track to be merged with at most one track, and (c) spatiotemporally neighboring tracks to be associated with neighboring candidate tracks. We encode these constraints into a discrete Markov Random Field (MRF), and compute a locally optimal linkage $\mathcal{L}$ of candidate tracks to query tracks. This linkage directly determines the resulting long-range tracks.

The MRF is formulated as follows (Fig. 2). Each query track τ_i is represented by a node in the graph, whose unknown state l_i is the index of a track to be linked to, and its candidate tracks form the state space for that node (the state space is likely to vary for different nodes in the graph). Since we do not wish to link a track at any cost, we present an additional state to each node with predefined cost δ (parameter), that will indicate that the corresponding track is deemed terminated. We model the compatibility of τ_i and a candidate track τ_j using unary potentials (local evidences) $\phi_i(l_i = j)$. This term will favor candidate tracks which follow visually-similar features and share common motion characteristics with τ_i. We then connect τ_i's node with nodes of other query tracks, τ_j, which reside in its spatiotemporal vicinity, and define the pairwise potentials $\psi_{ij}(l_i, l_j)$. These terms will assist in cases of tracking ambiguities and occlusion handling. Finally, an exclusion term, $\xi(\mathcal{L})$, is added, corresponding to an additional factor node in the graph [4]. This term enforces the linkage to be injective from queries to candidates. The probability of linkage $\mathcal{L}$ is then given by

$$P(\mathcal{L}) \propto \xi(\mathcal{L}) \prod_i \phi_i(l_i) \prod_{i,j \in \mathcal{N}(i)} \psi_{ij}(l_i, l_j), \tag{1}$$

where $\mathcal{N}(i)$ is the spatiotemporal neighborhood of track τ_i.

Track Compatibility. Let τ_i, τ_j be a query and candidate tracks, respectively ($t_i^{end} < t_j^{start}$). We factorize ϕ_i into three components: (a) appearance similarity, ϕ_i^a, (b) motion similarity, ϕ_i^m, and (c) a prior on the feature's motion while unobserved (occluded), ϕ_i^p, such that $\phi_i(l_i) = \phi_i^a(l_i)\phi_i^m(l_i)\phi_i^p(l_i)$.

We describe track τ_i's appearance at its termination time t_i^{end}, denoted $\tilde{\mathbf{s}}_i$, based on image features along the track. For each track point, we compute the SIFT descriptor in multiple scales and define the track's visual descriptor as a weighted average of the point descriptors along its last n_a frames

$$\tilde{\mathbf{s}}_i = \frac{1}{Z} \sum_{k=0}^{n_a-1} S_i(t_i^{end} - k) w_o(t_i^{end} - k) w_t(k) \tag{2}$$

where $S_i(t)$ is the track's SIFT descriptor at time t, $w_o(t)$ is an outlier weight, measuring how well $S_i(t)$ fits the track's general appearance, $w_t(k)$ is a time-decaying weight, and $Z = \sum_{k=0}^{n_a-1} w_o(t_i^{end} - k) w_t(k)$. $\tilde{\mathbf{s}}_j$, for candidate track τ_j, is symmetrically defined, considering its first n_a frames starting from t_j^{start}.

To measure appearance outliers, we first fit a Gaussian distribution G_i to the SIFT descriptors of the entire track τ_i, and set $w_o(t) = G_i(S_i(t))$. We use exponentially decaying weight $w_t(k) = \alpha_a^k, 0 < \alpha_a < 1$ (we will shortly specify the parameters we use). The appearance similarity is then defined in terms of the visual descriptors of the two tracks,

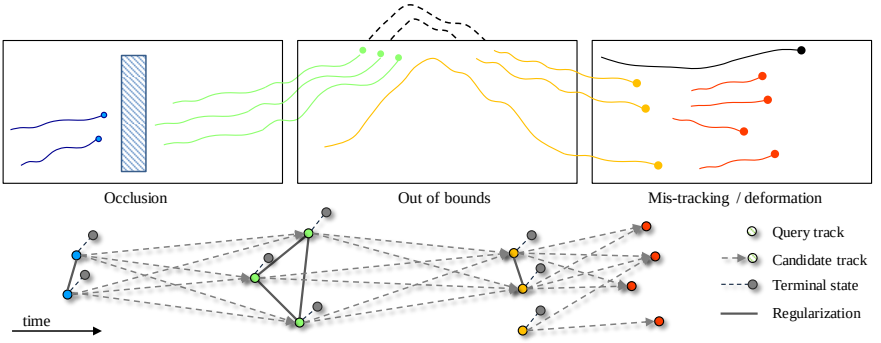


Figure 2: **The graphical model**, illustrated for common scenarios of track intermittence (top). Each track is represented by a node in graph and its state space (dashed lines) is comprised of its candidate tracks and an additional terminal state. Nearby tracks are connected by edges to regularize the linkage.

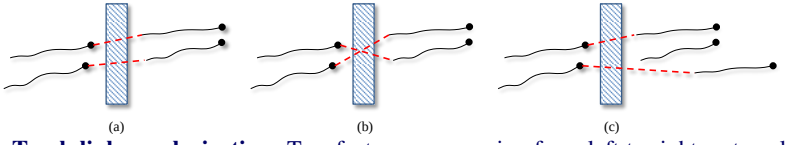


Figure 3: **Track link regularization**. Two features are moving from left to right, get occluded, and reappear from the right side of the occluder. (a-c) Assuming appearance and motion are similar in all cases, (a) is the link that will result in the highest (best) pairwise linking potential ψ_{ij}.

$$\phi_i^a(l_i = j) = \exp\left(-\frac{1}{\sigma_a^2}\left\|\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j\right\|_{1,d}\right), \tag{3}$$

where we use truncated L_1 norm $\|z\|_{1,d} = \min\left(\sum|z_i|, d\right)$ to account for appearance variation.

We similarly (and symmetrically for τ_j) estimate τ_i's velocity at its termination point as $\tilde{\mathbf{v}}_i = \frac{1}{Z}\sum_{k=0}^{n_v-1}\mathbf{v}_i(t_i^{end}-k)w_t(k)$, where $\mathbf{v}_i(t)$ is the observed velocity of τ_i at time t, and $w_t(k)$ is defined above. We then express the motion similarity between τ_i and τ_j with respect to their estimated end and start velocities, respectively,

$$\phi_i^m(l_i = j) = \exp\left(-\frac{1}{\sigma_m^2}\left\|\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j\right\|\right). \tag{4}$$

We also use a constant motion model for predicting the track's position while occluded,

$$\phi_i^p(l_i = j) = \exp\left(-\frac{1}{\sigma_p^2}\left\|\mathbf{x}_i(t_i^{end}) - \mathbf{x}_j(t_j^{start}) + (t_j^{start} - t_i^{end})\tilde{\mathbf{v}}_i\right\|\right). \tag{5}$$

This term will be typically assigned lower weight (larger σ_p^2), but we found it useful when points are occluded for extended periods. It can also be replaced with other motion models.

Link Regularization. We define the compatibility between a pair of query tracks as

$$\psi_{ij}(l_i = q, l_j = r) = \exp\left(-\frac{1}{\sigma_r^2}\left\|\mathbf{u}_{iq} - \mathbf{u}_{jr}\right\|\right), \tag{6}$$

where $\mathbf{u}_{ij} = \mathbf{x}_j(t_j^{start}) - \mathbf{x}_i(t_i^{end})$ is the spatiotemporal vector connecting the end of track τ_i with the beginning of track τ_j. This enforces neighboring query tracks to be linked to spatiotemporally close candidate tracks, and also penalizes links that cross trajectories behind occluders (Fig. 3).

Inference. we use loopy belief propagation to maximize Eq. 1 (local maximum is achieved). We fix $n_a = 20$, $n_v = 7$, and $\alpha_a = 0.4$. For efficiency, we prune the candidates for each query track and consider only the top K matches based on the track compatibility term, ϕ_i. We used $K = 100$ in our experiments (we discuss the effect of parameters on the algorithm in Sect. 4).

The free parameters of the algorithm are $\sigma_a, \sigma_m, \sigma_p, \sigma_r$, and δ, which we tuned manually on the sequences reported in Sect. 4. δ can be used to control the confidence level in which we allow the algorithm to operate. Larger value will restrict the algorithm to link tracks with higher certainty. We use the approximation in [4] to handle the exclusion term (refer to their paper, Sect. 3.3 and 3.4, for the message update equations).

3.3 Dynamic Scenes

In videos with moving camera, it is imperative to separate foreground (objects) from background (camera) motion, as camera pans and jitters may introduce arbitrary motions to the video that are difficult to model. We developed a simple motion-based stabilization algorithm that estimates affine camera motion using only the available tracklets. We found this algorithm to perform well and use it in all our experiments, however any stabilization algorithm can be used. The initial tracklets are first rectified (Fig. 5) and the algorithm continues from Sect. 3.2. We briefly review our stabilization algorithm in the supplemental material.

4 Experimental Results

We evaluated the algorithm on a set of synthetic and natural videos, and tested its applicability to human action recognition. The parameters were fixed for all the experiments to $\sigma_a = 40, \sigma_m = 6, \sigma_p = 12, \sigma_r = 25, \delta = 0.2$. A spatiotemporal radius of size 15 was used as the local neighborhood of each track for regularization. Importantly, small variations in K (*e.g.* 500, 1000) produced only marginal improvement in the results. The processing times on all the videos we tested were less than a minute (excluding the initial tracklets computation, which took $5-15$ minutes per video using the author's binary available online), on a 6-core Intel Xeon X5690 CPU with 32 GB RAM, and using our distributed C++ implementation. All the sequences and run times are available in the supplementary material.

In Fig. 4 we visualize the resulting motion tracks for a synthetic sequence (*car*; see below) and known computer vision sequences (*flowerGarden*, *sprites*). In all cases, the algorithm manages to link tracks of occluded features (see *e.g.* tracks on the car, the left house in *flowerGarden*, and the faces and background in *sprites*). Features on the leaves and branches of the tree in *flowerGarden*, which are not originally tracked continuously, are also properly linked in the result. Fig. 5 shows our result on a challenging natural video with rapid camera motion (*cheetah*). Albeit the visual abstractions from shadows, occlusions and motion blur, the algorithm managed to produce reasonable links, both on the cheetah, capturing its wobbly motion as it walks behind the tree, as well as on the background, where features enter and leave the frame due to camera pan. Note that the algorithm makes no use of any notion of an "object" (*e.g.* car), but is rather based solely on generic low-level cues.

Quantitative Analysis. One of the key challenges in devising long-range motion tracking algorithms is their evaluation. Existing datasets with ground-truth motions are available mostly for short sequences (2-10 frames) [1, 12], while to the best of our knowledge, no dataset or evaluation framework exists for dense long-range motion. [16] evaluated their particles by appending to the end of the video a temporally reversed copy of itself and measuring the error between the particle's start and end positions. This evaluation does not support intermittent tracks, as occluded particles cannot be re-correlated. Sundaram *et al.* [21] attempted to evaluate occlusion handling using the ground-truth annotations in [12], by checking if a track drifts between different motion segments. Such evaluation has no guarantee that the track will be associated with the same feature before and after occlusion.

We propose two complementary measures for long-range motion tracking. The first is based directly on our declared objective – to associate each scene feature with a single track. Given ground truth motion trajectories, we can thus consider the number of distinct tracks each scene point is associated with throughout the sequence as evaluation criteria. Towards

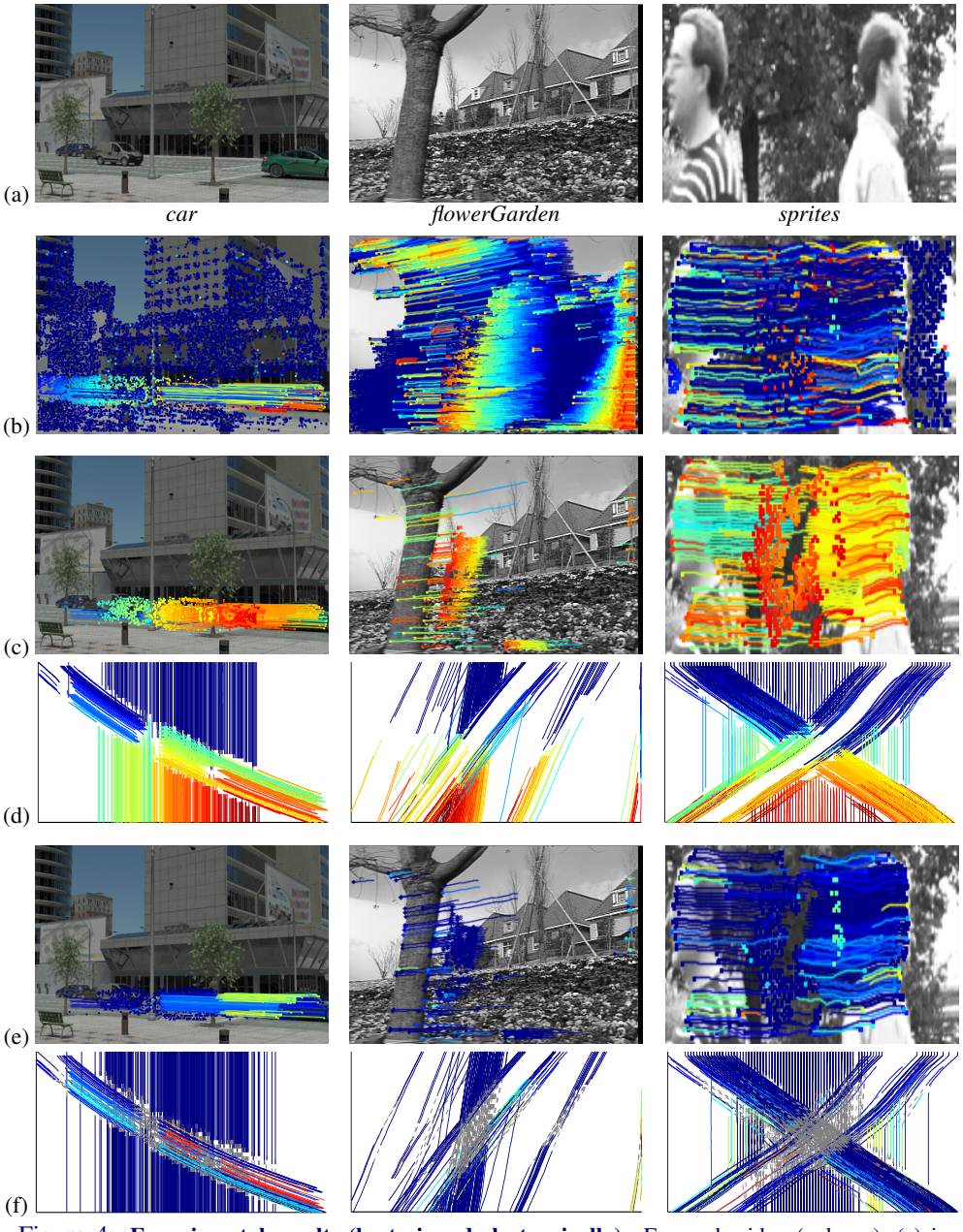
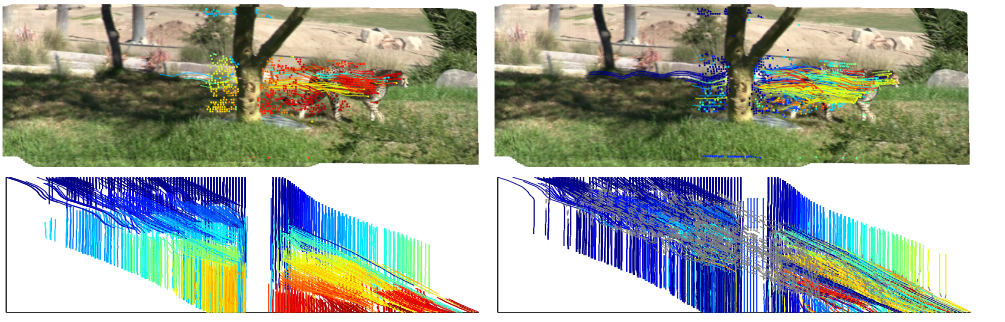


Figure 4: **Experimental results (best viewed electronically)**. For each video (column), (a) is a representative frame from the sequence, (b) are the resulting long-range motion tracks, (c) and (e) focus on the tracks involved in the linkage (tracks which are left unchanged are not shown), before (c) and after (e) they are linked. (d) and (f) show XT views of the tracks in (c) and (e), respectively, when plotted within the 3D video volume (time advancing downwards). The tracks are colored according to their initiation time, from blue (earlier in the video), to red (later in the video). Track links are shown as dashed gray lines in the spatiotemporal plots (d) and (f). For clarity of the visualizations, random samples ($25-50\%$) of the tracks are shown.



(a) Tracklets (state-of-the-art) (b) Long-range tracks (this paper)

Figure 5: **Result on a challenging natural sequence with moving camera** (*cheetah*). The initial tracklets (a) and resulting long-range tracks (b), are shown after stabilization (Sec. 3.3), for tracklets chosen to be linked by the algorithm (unmodified tracklets are not shown). The bottom plots show spatiotemporal XT slices of the corresponding tracks, similar to Fig. 4.

	PV	LDOF	PV + LR	LDOF + LR
r_{obj}	2.58	1.56	1.85	1.23

Table 1: **Quantitative evaluation using ground-truth motion trajectories** (*car*). The methods tested, from left to right: PV [16], LDOF [21], our long-range motion algorithm (LR) using PV as tracklets, and using LDOF as tracklets. These scores read, for example, "PV associated each scene point with 2.58 tracks on average throughout the video".

this end, we produced a synthetic photo-realistic simulation of an urban environment (*car*; Fig. 4) using the virtual city of [2]. We recorded the ground-truth motion from the renderer, and used it to compute ground-truth trajectories – the true 2D trajectories of 3D points in the scene. We define a point $\mathbf{y} = (x, y)$ in frame t to be associated with track τ, if the distance of the point to the track in that frame, $\|\mathbf{x}(t) - \mathbf{y}\|$, is sufficiently small, typically less than a quarter of a pixel. We compute the score of a tracking result, $r_{obj}(\Omega)$, by summing the number of tracks associated with each point, for all points in the sequence. We normalize the score by the number of points which are covered by tracks, to correct the bias towards a sparse solution, as the spatial density of the representation is not our focus in this work.

The results are summarized in Table 1, using tracks produced by PV, LDOF, and our algorithm, when using each of their results as initialization. Our algorithm significantly improves each algorithm separately and achieves the best score, 1.23, using the LDOF tracklets, with over 53% improvement over PV, and 22% improvement over LDOF.

The second measure takes into account the number of tracks initiated over time. Specifically, we compute the ratio $r(t) = \frac{\#\text{ tracks starting at frame } t}{\#\text{ tracks in frame } t}$, which we call the tracks' *refresh number*. In Fig. 6 we plot the refresh number for the aforementioned sequences, clearly showing that our algorithm initializes less tracks over time, utilizing existing tracks rather than creating new ones.

Action Recognition. What is long-range motion ultimately useful for? Since this representation captures better the motion in the scene, it should facilitate description, modeling and analysis of different types of motions. Here we describe preliminary experiments with one particular such task – human action recognition.

Previous work on action recognition combined image intensity statistics with motion statistics based on optical flow, either at each pixel location or along motion trajectories [20, 22]. In contrast, we are interested in descriptors based solely on the motion structures of tracks. Moreover, to leverage the long-range representation we also consider the long-term temporal characteristics of the motion (causality) that is discarded in previous bag-of-words approaches (Fig. 7(a)).

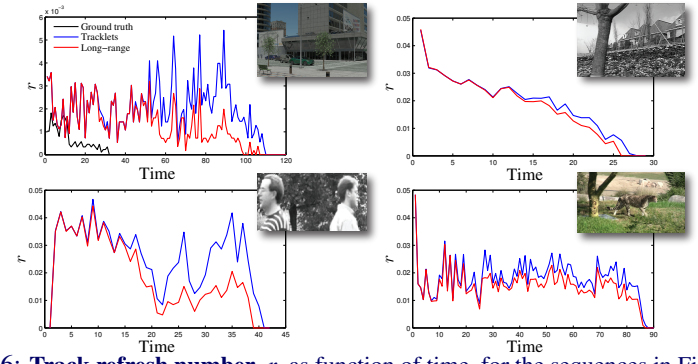


Figure 6: **Track refresh number**, r, as function of time, for the sequences in Fig. 4 and 5.

We used the KTH human action database [17] consisting of six human actions performed by 25 subjects, commonly used in this domain, and extracted long-range motion tracks as described above (using the same parameters for the algorithm). We divide each long-range track into $n_\sigma \times n_\sigma \times n_\tau$ spatiotemporal volumes (we used $n_\sigma = 7, n_\tau = 5$), and compute histograms of the velocities of tracks passing within each volume. The descriptor of each track is then defined as the concatenation of those motion histograms (Fig. 7(a)). To normalize for tracks of different lengths, we quantize each track to a fixed number of spatiotemporal cells relative to its length (5 cells in our implementation) and sum-up the histograms in each cell. Notice that the temporal extent of those cells is dependent on the length of the track – cells of longer tracks will be temporally longer than those of shorter tracks. This essentially captures motion structures at different *temporal scales*.

We then construct a codebook by clustering a random set of descriptors using K-means, and define the descriptor of each video as the histogram of the assignments of its motion track descriptors to the codebook vocabulary. For classification we use a non-linear SVM with χ^2-kernel (see [22] for the details). As in [17], we divided the data per person, such that 16 persons are used as training set and 9 persons are used for testing. We train the classifier on the training set and report recognition results on the test set in Fig. 7.

The overall recognition rate in our preliminary experiment on this dataset, 76.5% (Fig. 7(b)), does not reach the state-of-the-art using spatiotemporal features, 86.8% [20]. However, while the best-performing methods use several types of features (including optical flow statistics), our descriptor is based solely on the tracks' motions. Our algorithm outperforms [17], 71.7%, which is based on spatiotemporal image structures. Fig. 7(c) shows that the long-range trajectories outperformed the initial tracklets on almost all action classes, demonstrating the potential of a long-range motion representation for action recognition.

5 Conclusion

We have presented an algorithm for obtaining long-range motion trajectories in video sequences. In order to properly handle the disappearance and reappearance of features, distant frames in the sequence need to be correlated. Following a divide-and-conquer paradigm, we build upon state-of-the-art feature trackers to produce an initial set of short accurate track estimates, and then link the tracks to form long trajectories such that each scene feature is associated with a single track throughout the video with high probability. We formulate track linking over the entire sequence as a combinatorial association problem based on appearance and motion cues, as well as track inter-relations. This both utilizes information from different parts of the sequence and helps resolve link ambiguities. We demonstrated encouraging results on both synthetic and natural sequences. For all sequences we tested, the algorithm manages to improve the state-of-the-art results. We also showed applications of the long-range trajectory representation to human action recognition.

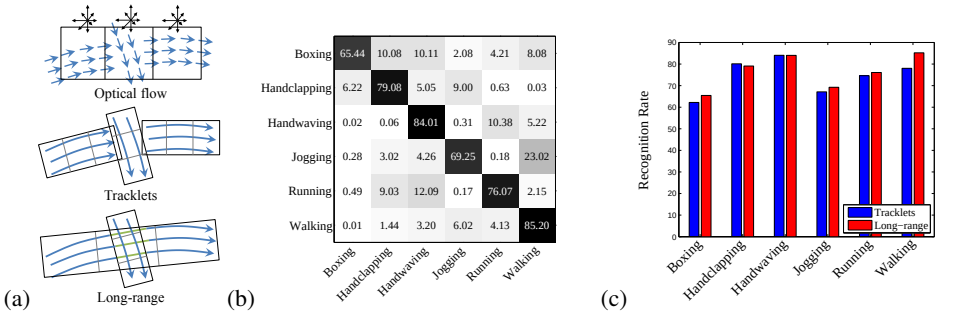


Figure 7: **Recognition results on the KTH human action database.** (a) Our motion descriptor (bottom) in comparison to existing motion descriptors based on optical flow (top) and short-range tracks (middle). (b) The confusion matrix using the long-range trajectories produced by our algorithm. (c) Comparison of the recognition rates when using tracklets (used as initialization for our algorithm) versus the resulting long-range trajectories.

Acknowledgments

We would like to thank Rick Szeliski for helpful discussions. This material is based upon work supported by the National Science Foundation under Grant No. CGV 1111415 and by an NVDIA Fellowship to M. Rubinstein.

References

[1] S. Baker, D. Scharstein, JP Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[2] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE, 2009.

[3] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[4] T.S. Cho, S. Avidan, and W.T. Freeman. The patch transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(8):1489–1501, 2010.

[5] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282, 2008.

[6] W. Ge and R.T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *BMVC*, volume 96, 2008.

[7] B. K. P. Horn and B. G. Schunck. Determing optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] H. Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[9] Biliana Kaneva, Antonio Torralba, and William T. Freeman. Evaluating image feaures using a photorealistic virtual world. In *IEEE International Conference on Computer Vision*, pages 2282–2289, 2011.

[10] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1683–1698, 2008.

[11] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2953–2960. IEEE, 2009.

[12] C. Liu, W.T. Freeman, E.H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[13] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[14] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking-linking identities using bayesian network inference. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2187–2194. IEEE, 2006.

[15] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: a new representation for video editing. In *ACM SIGGRAPH 2008 papers*, pages 1–11. ACM, 2008.

[16] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2195–2202. IEEE, 2006.

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[18] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, June 1994.

[19] B. Song, T.Y. Jeng, E. Staudt, and A. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. *European Conference on Computer Vision (ECCV)*.

[20] Ju Sun, Yadong Mu, Shuicheng Yan, and Loong Fah Cheong. Activity recognition using dense long-duration trajectories. In *ICME*, pages 322–327, 2010.

[21] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. *Computer Vision–ECCV 2010*, pages 438–451, 2010.

[22] H. Wang, A. Klaser, C. Schmid, and C.L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2011.